CE Linux Forum Power Management Enhancements
by Geoff Levand

A new power management sub-system was introduced in the Linux-2.6 kernel. This new sub-system, which uses the LDM (Linux Device Model), views the system as a hierarchy of buses and devices, and manages the system accordingly. Although the basic power management functionality of this new sub-system is working as of Linux-2.6.11, development is still ongoing, and a lack of support for non-PC platforms and other key features makes it troublesome for CE product developers.

The power management enhancements outlined in this article were born from this lack of support. These enhancements need to be viewed as proof-of-concept, not finished works. The platform support patches will be promoted to platform maintainers, and the safe-suspend.patch, which is of interest to Linux users in general, will be promoted to kernel developers. Other features outlined here will not be of benefit to the general Linux Community, and may only find support in a small group of CE device makers.

Summary of Patches

*** pm-reorder-freeze.patch**

This patch changes the order in which the kernel attempts to freeze system threads on suspend and disables kernel preemption while thawing threads on resume.

During system suspend the original kernel routines used the for_each_process() macro to iterate over the kernel's task list starting with init_task. If a thread was found to be freezable, the power management sub-system would put the thread into a frozen state, irrespective of the type of thread. The patched routine does this iteration twice, the first time testing if the thread is a user mode thread. In this way an attempt is made to freeze user mode threads before an attempt is made to freeze kernel threads.

During system resume the original kernel routines took no precaution in the ordering of thread thaw and execution. The patched routines prevent execution of all user mode threads until after all threads are thawed by disabling kernel preemption during the thaw operation.

*** pm-port-busy.patch**

This patch adds four new routines, serial8250_suspend_port_busy(), serial8250_resume_port_busy(), uart_suspend_port_busy(), and uart_resume_port_busy(). These routines provide non-sleeping versions of existing kernel power management routines. The non-sleeping versions can be used by architecture specific suspend and resume routines to maximize the availability of the serial console for system logging during a suspend/resume operation.

These routines are nearly identical to the existing kernel power management routines serial8250_suspend_port(), serial8250_resume_port(), uart_suspend_port(), and uart_resume_port(), except that no semaphores are used, and msleep() loops are replaced with TASK_UNINTERRUPTIBLE timeouts in I/O completion waits.

*** pm-on-ebony.patch**

This patch provides power management support for the IBM PPC440GP Ebony Reference Platform. The main portion of the patch implements the platform specific pm_ops structure required by the kernel power management sub-system. The current implementation only supports suspend-to-memory (PM_SUSPEND_MEM), though unpublished suspend-to-disk work has been started.

This implementation arranges for the U44 switch on the Ebony platform, connected to the SMI interrupt handler, to be used as a system resume trigger.

*** safe-suspend.patch**

This patch provides a new kernel configuration option 'Mark FS clean on Suspend'. This feature remounts all filesystems as read-only before suspending the system, and as read-write on resume. This feature assures filesystem integrity in the case of accidentally exchanging batteries with a notebook computer in standby state, or powering down a desktop PC while in standby state. Without this feature enabled, filesystems will require repair on restart.

This feature works by adding calls to suspend_remount() and resume_remount() at points in the suspend/resume sequence when all system threads are frozen. This is done in the existing suspend_prepare() and suspend_finish() power management routines.

*** fast-clean-shutdown.patch**

This patch provides a new kernel configuration option 'Fast & Clean Shutdown'.

A fast system shutdown is accomplished by simply freezing all processes, remounting all filesystems as read-only, closing all TCP connections, then powering down. The suitability of this feature is very dependent on the particular device configuration, but may be of use for some CE devices.

*** deferred-resume.patch**

This patch provides a new kernel configuration option 'Deferred resume'.

This feature creates a dedicated system task to execute resume operations for tasks not in the process group of the suspend initiator task. With this feature, execution of the suspend initiator will return early, and execute in parallel with the resume of other tasks. The suspend initiator task can be programmed to take advantage of this feature, for example, to display a splash screen early in the system resume.

How to Prepare Kernel Sources

Download the archive file 'celf-pm-patches-interface.tar.bz2' from the CE Linux Forum Patch Archive:

http://tree.celinuxforum.org/CelfPubWiki/PatchArchive

The archive contains a directory 'patches', setup for the patch management utility quilt. Use the command 'quilt setup' if you have this utility available on your system. Otherwise, the patches can be applied manually in the order specified in the quilt series file 'patches/series' as in the commands below.

```
$ cd linux-2.6.11
$ tar -xjf celf-pm-patches.tar.bz2
$ cat patches/series | egrep -v '(^#|^$)' | sed 's/\([^ ].\) $/\1/' \
| xargs -i cat patches/{} | patch -p1
```

With the patches applied, several new kernel configuration options will be available in the Power Management sub-menu:

```
*
* Power management options
*
Power Management support (PM) [Y/n/?] y
Power Management Debug Support (PM_DEBUG) [Y/n/?] y
Mark FS clean on Suspend (EXPERIMENTAL) (SAFE_SUSPEND) [Y/n/?] y
Fast & Clean Shutdown (EXPERIMENTAL) (FAST_CLEAN_SHUTDOWN) [Y/n/?] y
Deferred resume (DEFERRED_RESUME) [Y/n/?] y
```

How to Exercise New Features

To exercise the features covered in this article, a properly configured kernel, as described in the tests below, needs to be built and installed to the system. The system must also have an unused mass storage partition. Some systems may need minor preparation to allow power management operations.

* Preparation of sys Filesystem

The new power management sub-system exports an interface via the sys filesystem. If needed, prepare the system as follows:

```
# mkdir -p /sys
# mount -t sysfs sysfs /sys
```

* Filesystem I/O Helper

To exercise the enhancements to filesystem robustness, a helper script is used that assures pending filesystem operations are present during power management suspend. This example uses the unused partition '/dev/hde3'. The helper script is available in the patches directory 'patches/pm/test-fs.sh'.

To prepare for a test run, create a fresh ext2 filesystem on the unused partition, and mount the new filesystem in a convenient location as shown below. The first argument to the script indicates the working directory for the script.

```
# umount /dev/hde3
# mkfs.ext2 /dev/hde3
# mount /dev/hde3 /mnt
# ./test-fs.sh /mnt/test

Doing fs test with /mnt/test.
== do_test 2 1
== MKDIR: D_0 D_10
-:D_0...
== do_test 2 2
== MKDIR: D_0 D_10
-:D_0...
-:D_10...
```

*** Base System Test**

This test displays the behavior of the base kernel without the CELF enhancements. Configure, build and install a kernel with the following options:

```
*
* Power management options
*
Power Management support (PM) [Y/n/?] y
Power Management Debug Support (PM_DEBUG) [Y/n/?] y
Mark FS clean on Suspend (EXPERIMENTAL) (SAFE_SUSPEND) [Y/n/?] n
Fast & Clean Shutdown (EXPERIMENTAL) (FAST_CLEAN_SHUTDOWN) [Y/n/?] n
Deferred resume (DEFERRED_RESUME) [Y/n/?] n
```

Also install the helper script 'test-fs.sh'. To assure the proper kernel configuration options were set, list the contents of the '/sys/power/' directory.

```
# ls /sys/power/
state
```

Next, start the helper script. This can be done as a background task on the system console, but it is recommended to use a console separate from the system console so that the outputs from the helper script and power management sub-system are not interleaved. It is also recommended to create a fresh filesystem with 'mkfs.ext2' every time the helper script is run.

Now initiate a power management suspend operation by write the string 'mem' to '/sys/power/state', then initiate a power management resume. The system should resume with all tasks back up as expected.

```
/ # echo mem > /sys/power/state
Stopping tasks: ==============|
066145.077ms:WAKEUP
PM: Finishing up.
Restarting tasks... done
```

Next, test for filesytem corruption on incomplete resume. Initiate a power management suspend operation.

```
# echo mem > /sys/power/state
Stopping tasks: ================|
```

At this point the system should enter into the suspend state. Since the helper script was executing, there were pending filesystem operations at the time of suspend. Cycle the system power to initiate a cold system boot. On restart, a filesystem check will show a corrupted filesystem. This example uses the ext2 filesystem, but the FAT family of filesytems, common on CE platforms, are also effected.

```
# e2fsck -n /dev/hde3
e2fsck 1.27 (8-Mar-2002)
/dev/hde3 was not cleanly unmounted, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Entry 'test' in / (2) has an incorrect filetype (was 2, should be 0).
Fix? no
```

```
Pass 3: Checking directory connectivity
Unconnected directory inode 162241 (/???)
Connect to /lost+found? no
'..' in ... (162241) is / (2), should be <The NULL inode> (0).
Fix? no
Pass 4: Checking reference counts
Inode 162241 ref count is 3, should be 2.  Fix? no
Pass 5: Checking group summary information
/dev/hde3: ********** WARNING: Filesystem still has errors **********
/dev/hde3: 111/486720 files (4.5% non-contiguous), 40227/972468 blocks
```

*** Test of safe-suspend**

This test displays the behavior of the safe-suspend enhancement. Configure, build and install a kernel with the following options:

```
*
* Power management options
*
Power Management support (PM) [Y/n/?] y
Power Management Debug Support (PM_DEBUG) [Y/n/?] y
Mark FS clean on Suspend (EXPERIMENTAL) (SAFE_SUSPEND) [Y/n/?] y
Fast & Clean Shutdown (EXPERIMENTAL) (FAST_CLEAN_SHUTDOWN) [Y/n/?] y
Deferred resume (DEFERRED_RESUME) [Y/n/?] y
```

Note that this same option configuration is used for all remaining tests, so the same kernel can be used. List the contents of the '/sys/power/' directory to check the configuration.

```
# ls /sys/power/
deferred_resume fast_clean_shutdown state
```

Start the helper script and then initiate a power management suspend operation.

```
# echo mem > /sys/power/state
Stopping tasks: ==============|
suspend_remount:hde3
```

Cycle the system power to initiate a cold system boot. On restart, a filesystem check will show the filesystem is clean.

```
#  e2fsck -n /dev/hde3
e2fsck 1.27 (8-Mar-2002)
/dev/hde3: clean, 114/486720 files, 41252/972468 blocks
```

*** Test of fast-clean-shutdown**

This test displays the behavior of the fast-clean-shutdown enhancement. Configure, build and install a kernel with the following options:

```
*
* Power management options
*
Power Management support (PM) [Y/n/?] y
Power Management Debug Support (PM_DEBUG) [Y/n/?] y
Mark FS clean on Suspend (EXPERIMENTAL) (SAFE_SUSPEND) [Y/n/?] y
Fast & Clean Shutdown (EXPERIMENTAL) (FAST_CLEAN_SHUTDOWN) [Y/n/?] y
```

```
Deferred resume (DEFERRED_RESUME) [Y/n/?] y
```

A check can be made to test if this feature is enabled by reading from 'sys/power/fast_clean_shutdown'.

```
# cat sys/power/fast_clean_shutdown
enabled
```

Start the helper script and then initiate a fast clean shutdown by writing '1' to 'sys/power/fast_clean_shutdown'.

```
# echo 1 > sys/power/fast_clean_shutdown
Stopping tasks: ================|
suspend_remount:hde3
Shutdown: hde
```

If running a telnet session, an indicator of a clean tcp close may be seen at the telnet client:

```
Doing fs test with /mnt/test.
== do_test 2 1
== MKDIR: D_0 D_10
-:D_0..Connection closed by foreign host.
```

After restarting the system a filesystem check will show the filesystem is clean.

```
#  e2fsck -n /dev/hde3
e2fsck 1.27 (8-Mar-2002)
/dev/hde3: clean, 114/486720 files, 41252/972468 blocks
```

*** Test of deferred-resume**

This test displays the behavior of the deferred-resume enhancement. Configure, build and install a kernel with the following options:

```
*
* Power management options
*
Power Management support (PM) [Y/n/?] y
Power Management Debug Support (PM_DEBUG) [Y/n/?] y
Mark FS clean on Suspend (EXPERIMENTAL) (SAFE_SUSPEND) [Y/n/?] y
Fast & Clean Shutdown (EXPERIMENTAL) (FAST_CLEAN_SHUTDOWN) [Y/n/?] y
Deferred resume (DEFERRED_RESUME) [Y/n/?] y
```

A check can be made to test if this feature is enabled by reading from 'sys/power/deferred_resume'. The default state is disabled. To change the state write 'enable' or 'disable' to 'sys/power/deferred_resume'.

```
# cat /sys/power/deferred_resume
disable
# echo enable > /sys/power/deferred_resume
# cat /sys/power/deferred_resume
enable
```

The process list shows the dedicated resume task 'kresume'.

```
# ps
```

```
PID TTY          TIME CMD
1 ?        00:00:03 init
2 ?        00:00:00 ksoftirqd/0
3 ?        00:00:03 events/0
4 ?        00:00:00 khelper
5 ?        00:00:00 kthread
6 ?        00:00:04 kblockd/0
8 ?        00:00:00 kresume
7 ?        00:00:00 khubd
9 ?        00:00:00 pdflush
10 ?        00:02:58 pdflush
12 ?        00:00:00 aio/0
11 ?        00:00:06 kswapd0
13 ?        00:00:00 mtdblockd
14 ?        00:00:00 rpciod/0
23 ?        00:00:01 ash
24 ?        00:00:00 telnetd
14028 ?        00:00:00 ps
```

Initiate a power management suspend operation, and then a power management resume. The system should resume with all tasks back up as expected, and as seen in the output below, execution of the suspend initiator task, indicated by the message 'PM: Return without finishing up' occurs before other system tasks, indicated by the 'Restarting tasks...' message.

```
/ # echo mem > /sys/power/state
Stopping tasks: ==============|
suspend_remount:hde3
204213.223ms:WAKEUP
204236.799ms:[7921]:pm_sem:0:<echo>:Back
Thaw pgid(0): Strange, [0] swapper not stopped
204338.762ms:[7921]:pm_sem:0:<echo>:woke up minimal threads
PM: Return without finishing up.
204445.158ms:[7921]:pm_sem:0:<echo>:Return without finishing up
204517.999ms:[008]:pm_sem:0:<kresume>:begin deferred finishing up
PM(deferred): Finishing up.
/ # resume_remount: hde3
Restarting tasks...<6> [1] init not stopped
[23] ash not stopped
done
206639.443ms:[008]:pm_sem:0:<kresume>:end deferred finishing up
206751.273ms:[008]:pm_sem:1:<kresume>:waiting completion
```

Summary

The improvements outlined here are just a few of many that could be done to enhance the Linux kernel for CE product use. Also, other areas of kernel power management, for instance, suspend-to-disk, are under active development and have a need of improvement for use in CE products.

More information regarding the new Linux-2.6 power management sub-system can be found in these references:

http://tree.celinuxforum.org/CelfPubWiki/PmSubSystem
http://archive.linuxsymposium.org/ols2003/Proceedings/All-Reprints/Reprint-Mochel-OLS2003.pdf

Information about suspend-to-disk support is available at the Software Suspend2 Web site:

  http://www.suspend2.net